Oracle Day 6 – Joins in Oracle

Note: Please watch my YouTube sessions to better understand the descriptions and queries below

NiC IT Academy YouTube Videos for reference

🔵 Oracle SQL Tutorial - English

https://youtube.com/playlist?list=PLsphD3EpR7F9mmtY2jBt_O8Q9XmvrhQEF

🌕 Oracle SQL - தமிழில்

https://youtube.com/playlist?list=PLsphD3EpR7F-u4Jjp_3fYgLSsKwPPTEH4

📌 Oracle SQL Day wise Video: ENGLISH

Oracle SQL Day 1 – Introduction to Oracle - https://youtu.be/hLnKjYGr730

Oracle SQL Day 2 – SQL Types DDL, DML, DRL, DCL, TCL - https://youtu.be/XpgjXvnfZec

Oracle SQL Day 3 – Constraints in Oracle - https://youtu.be/TmYqeFfHyyc

Oracle SQL Day 4 – SELECT Statements in Oracle - https://youtu.be/tYQfBgUCpoI

Oracle SQL Day 5 – Single Row Functions in Oracle - https://youtu.be/4qJJxQuHLC4

Oracle SQL Day 6 – Joins in Oracle - https://youtu.be/CkaqluC2afE

Oracle SQL Day 7 – Aggregate Functions in Oracle - https://youtu.be/BSiCWzj-py8

Oracle SQL Day 8 – Sub Queries in Oracle - https://youtu.be/KtUCyG2cZe4

Oracle SQL Day 9 – SET Operators in Oracle - https://youtu.be/B0JbGbWsEIA

Oracle SQL Day 10 – Analytical Functions in Oracle - https://youtu.be/gRC3ndWLsoo

Oracle SQL Day 11 - Views in Oracle - https://youtu.be/m8a1UtOmd5k

Oracle SQL Day 12 - Indexes in Oracle - https://youtu.be/reL2O-kvNxc

Oracle SQL Day 13 - Regular Expression - https://youtu.be/k_Eo08vLPhU

Customer c

cust_id	cust_name	mob_no	email	country_id
1000	Rakesh	8989898989	Rakesh@gmail.com	200
1001	Arun	8989898990	Arun@gmail.com	202
1002	Maxwell	8989898991	Maxwell@gmail.com	204
1003	Shankar	8989898992	Shankar@gmail.com	200
1004	Sree	8989898993	Sree@gmail.com	204
1005	Sam	8989898994	Sam@gmail.com	205
1006	Radha	8989898995	Radha@gmail.com	203
1007	Senthil	8989898996	Senthil@gmail.com	200

Country r

Country_id	Country_name
200	INDIA
201	CHINA
202	USA
203	SINGAPORE
204	UK

Inner Join

cust_id	cust_name	mob_no	email	country_id	Country_name
1000	Rakesh	8989898989	Rakesh@gmail.com	200	INDIA
1001	Arun	8989898990	Arun@gmail.com	202	USA
1002	Maxwell	8989898991	Maxwell@gmail.com	204	UK
1003	Shankar	8989898992	Shankar@gmail.com	200	INDIA
1004	Sree	8989898993	Sree@gmail.com	204	UK
1006	Radha	8989898995	Radha@gmail.com	203	SINGAPORE
1007	Senthil	8989898996	Senthil@gmail.com	200	INDIA

Left Outer Join

cust_id	cust_name	mob_no	email	country_id	Country_name
1000	Rakesh	8989898989	Rakesh@gmail.com	200	INDIA
1001	Arun	8989898990	Arun@gmail.com	202	USA
1002	Maxwell	8989898991	Maxwell@gmail.com	204	UK
1003	Shankar	8989898992	Shankar@gmail.com	200	INDIA
1004	Sree	8989898993	Sree@gmail.com	204	UK

1005	Sam	8989898994	Sam@gmail.com	205	NULL
1006	Radha	8989898995	Radha@gmail.com	203	SINGAPORE
1007	Senthil	8989898996	Senthil@gmail.com	200	INDIA

Right Outer Join

cust_id	cust_name	mob_no	email	country_id	Country_name
1000	Rakesh	8989898989	Rakesh@gmail.com	200	INDIA
1001	Arun	8989898990	Arun@gmail.com	202	USA
1002	Maxwell	8989898991	Maxwell@gmail.com	204	UK
1003	Shankar	8989898992	Shankar@gmail.com	200	INDIA
1004	Sree	8989898993	Sree@gmail.com	204	UK
1006	Radha	8989898995	Radha@gmail.com	203	SINGAPORE
1007	Senthil	8989898996	Senthil@gmail.com	200	INDIA
NULL	NULL	NULL	NULL	201	CHINA

Full Outer Join

cust_id	cust_name	mob_no	email	country_id	Country_name
1000	Rakesh	8989898989	Rakesh@gmail.com	200	INDIA
1001	Arun	8989898990	Arun@gmail.com	202	USA
1002	Maxwell	8989898991	Maxwell@gmail.com	204	UK
1003	Shankar	8989898992	Shankar@gmail.com	200	INDIA
1004	Sree	8989898993	Sree@gmail.com	204	UK
1005	Sam	8989898994	Sam@gmail.com	205	NULL
1006	Radha	8989898995	Radha@gmail.com	203	SINGAPORE
1007	Senthil	8989898996	Senthil@gmail.com	200	INDIA
NULL	NULL	NULL	NULL	201	CHINA

Joins in Oracle:

============

Equi join - An equi join is a type of join that combines tables

based on matching values in specified columns. =

Non-Equi join - The nonequijoins is such a join which match column values from different tables based on an inequality

(instead of the equal sign like >, <, >=, <=) expression

4 Types of Equi join:

=================

Inner Join

Left Outer join

Right Outer Join

Full Outer Join

2 methods to write join query:

==============================

1. Implicit method
2. ANSI Method

Table scripts for practice:

****

```
create table customer

(

cust_id  number,

cust_name        varchar2(50),

mob_no           number(10),

email    varchar2(50),
country_id number(3)

);


Insert into customer (CUST_ID,CUST_NAME,MOB_NO,EMAIL,COUNTRY_ID) values
(1000,'Kannan',8989898989,'kannan@gmail.com',200);

Insert into customer (CUST_ID,CUST_NAME,MOB_NO,EMAIL,COUNTRY_ID) values
(1001,'Arun',8989898990,'arun@gmail.com',204);

Insert into customer (CUST_ID,CUST_NAME,MOB_NO,EMAIL,COUNTRY_ID) values
(1002,'Karthik',8989898991,'Karthik@gmail.com',202);

Insert into customer (CUST_ID,CUST_NAME,MOB_NO,EMAIL,COUNTRY_ID) values
(1003,'Shankar',8989898992,'shankar@gmail.com',203);

Insert into customer (CUST_ID,CUST_NAME,MOB_NO,EMAIL,COUNTRY_ID) values
(1004,'Sree',8989898993,'Sree@gmail.com',205);

Insert into customer (CUST_ID,CUST_NAME,MOB_NO,EMAIL,COUNTRY_ID) values
(1005,'Babu',8989898994,'Babu@gmail.com',200);

Insert into customer (CUST_ID,CUST_NAME,MOB_NO,EMAIL,COUNTRY_ID) values
(1006,'Radha',8989898995,'Radha@gmail.com',202);

Insert into customer (CUST_ID,CUST_NAME,MOB_NO,EMAIL,COUNTRY_ID) values
(1007,'Senthil',8989898996,'Senthil@gmail.com',200);



commit;



create table country

(

Country_id       number(3),

Country_name varchar2(50)

);


Insert into country (COUNTRY_ID,COUNTRY_NAME) values (200,'INDIA');

Insert into country (COUNTRY_ID,COUNTRY_NAME) values (201,'CHINA');
Insert into country (COUNTRY_ID,COUNTRY_NAME) values (202,'USA');

Insert into country (COUNTRY_ID,COUNTRY_NAME) values (203,'SINGAPORE');

Insert into country (COUNTRY_ID,COUNTRY_NAME) values (204,'UK');


commit;
```

==

Inner Join: -- implicit method

```
select
c.cust_id,
c.cust_name,
c.mob_no,
c.email,
c.country_id,
r.country_name
from customer c, country r
where c.country_id=r.country_id;
```

Inner Join: -- ANSI Method

```
select
c.cust_id,
c.cust_name,
c.mob_no,
c.email,
c.country_id,
r.country_name

from customer c inner join country r

on c.country_id=r.country_id;
```

****

Left Outer Join:

```
------------

  -- implicit



select

c.cust_id,

c.cust_name,

c.mob_no,

c.email,

c.country_id,

r.country_name

from customer c, country r

where c.country_id=r.country_id(+);
```

Left Outer Join:

```
------------

  -- ANSI



select

c.cust_id,
c.cust_name,

c.mob_no,

c.email,

c.country_id,

r.country_name

from customer c Left Outer join country r

on c.country_id=r.country_id;
```

****

Right Outer Join:

```
  -- implicit



select

c.cust_id,

c.cust_name,

c.mob_no,

c.email,

c.country_id,

r.country_name

from customer c, country r

where c.country_id(+)=r.country_id;
```

Right Outer Join:

-- ANSI

```
select
c.cust_id,
c.cust_name,
c.mob_no,
c.email,
c.country_id,
r.country_name
from customer c Right Outer join country r
on c.country_id=r.country_id;
```

****

Full Outer Join:

-- implicit method

```
select
c.cust_id,
c.cust_name,
c.mob_no,
c.email,
c.country_id,
r.country_name
from customer c, country r
where c.country_id=r.country_id(+)
union
```

```
select

c.cust_id,

c.cust_name,

c.mob_no,

c.email,

c.country_id,

r.country_name

from customer c, country r

where c.country_id(+)=r.country_id;
```

-- ANSI

```
select

c.cust_id,

c.cust_name,

c.mob_no,

c.email,

c.country_id,

r.country_name

from customer c Full Outer join country r

on c.country_id=r.country_id;
```

****

3 tables -- customer city country

```
select
c1.cust_id,
c1.cust_name,
c1.mob_no,
c1.email,
c1.city_id,
c2.city_name,
c2.country_id country_id,
c3.country_name
from customer c1,city c2, country c3
where c1.city_id=c2.city_id
and c2.country_id=c3.country_id;
```

```
select
c1.cust_id,
c1.cust_name,
c1.mob_no,
c1.email,
c1.city_id,
c2.city_name,
c2.country_id country_id,
c3.country_name
from customer c1 inner join city c2
on c1.city_id=c2.city_id
inner join country c3
```

```
on c2.country_id=c3.country_id;

***********************************************************

-- 2 tables join – HR Schema tables

select * from employees;

select * from departments;

select
e.employee_id,
e.first_name,
e.email,
e.SALARY,
e.department_id,
d.DEPARTMENT_NAME,
d.LOCATION_ID
from Employees e,departments d
where e.department_id=d.department_id;


select
e.employee_id,
e.first_name,
e.email,
e.SALARY,
e.department_id,
d.DEPARTMENT_NAME,
d.LOCATION_ID
```

```
from employees e inner join DEPARTMENTS d
on e.department_id=d.department_id;


-- Left Outer Join

select
e.employee_id,
e.first_name,
e.job_id,
e.salary,
e.department_id,
d.DEPARTMENT_NAME
from employees e, departments d
where e.department_id=d.department_id(+);


select
e.employee_id,
e.first_name,
e.job_id,
e.salary,
e.department_id,
d.DEPARTMENT_NAME
from employees e left join departments d
on e.department_id=d.department_id;

-----------------------------------------------------
```

```
-- Right outer join

select
e.employee_id,
e.first_name,
e.job_id,
e.salary,
d.department_id,
d.DEPARTMENT_NAME
from employees e, departments d
where e.department_id(+)=d.department_id;



select
e.employee_id,
e.first_name,
e.job_id,
e.salary,
e.department_id,
d.DEPARTMENT_NAME
from employees e right join departments d
on e.department_id=d.department_id;

-- Full Outer Join

select
e.employee_id,
e.first_name,
e.job_id,
e.salary,

e.department_id,

d.DEPARTMENT_NAME

from employees e full join departments d

on e.department_id=d.department_id;
```

****

select the departmets where no employees are working

```
select

d.department_id,

d.DEPARTMENT_NAME

from employees e, departments d

where e.department_id(+)=d.department_id and e.employee_id is null;
```

===

-- 3 tables join – HR Schema tables

```
select * from employees;

select * from departments;

select * from locations;


select

e.employee_id,

e.first_name,

e.job_id,

e.salary,
e.department_id,
d.DEPARTMENT_NAME,
d.location_id,
l.street_address,
l.city,
l.country_id
from employees e,departments d,locations l
where e.department_id=d.department_id
and d.location_id=l.location_id;


select
e.employee_id,
e.first_name,
e.job_id,
e.salary,
e.department_id,
d.DEPARTMENT_NAME,
d.location_id,
l.country_id
from employees e inner join departments d
on e.department_id=d.department_id
inner join locations l
on  d.location_id=l.location_id;


select
e.employee_id,
e.first_name,

e.job_id,

e.salary,

e.department_id,

d.DEPARTMENT_NAME,

d.location_id,

l.country_id,

c.COUNTRY_NAME

from employees e inner join departments d

on e.department_id=d.department_id

inner join locations l

on  d.location_id=l.location_id

inner join COUNTRIES c on l.COUNTRY_ID=c.COUNTRY_ID

where l.country_id='CA' and e.salary >10000;
```

----

4 tables join with sub query:

=============================

```
select

e.employee_id,

e.first_name,

e.job_id,

e.salary,

e.department_id,

d.DEPARTMENT_NAME,

d.location_id,

l.country_id,

c.country_name

from employees e inner join (select department_id,department_name,location_id from departments) d
```

```
on e.department_id=d.department_id
join locations l
on d.location_id=l.location_id
join countries c
on l.country_id=c.country_id where c.country_name='United Kingdom' and e.salary >10000;


-- 5 tables Implicit joins

select
e.employee_id,
e.first_name,
e.job_id,
e.salary,
case when salary >10000 then 'High salary' else 'Low Salary' end salary_status,
e.department_id,
d.DEPARTMENT_NAME,
d.location_id,
l.country_id,
c.country_name,
j.job_title
from employees e, (select department_id,department_name,location_id from departments) d, locations l, countries c,jobs j
where e.department_id=d.department_id
and d.location_id=l.location_id
and l.country_id=c.country_id
and e.job_id=j.job_id and l.country_id in ('UK','CA') and e.job_id='SA_REP';

-- 5 tables ANSI joins
```

```
select
e.employee_id,
e.first_name,
e.job_id,
e.salary,
case when salary >10000 then 'High salary' else 'Low Salary' end salary_status,
e.department_id,
d.DEPARTMENT_NAME,
d.location_id,
l.country_id,
c.country_name,
j.job_title
from employees e inner join (select department_id,department_name,location_id from departments) d
on e.department_id=d.department_id
join locations l
on d.location_id=l.location_id
join countries c
on l.country_id=c.country_id
join jobs j
on e.job_id=j.job_id
where l.country_id in ('UK','CA') and e.job_id='SA_REP';

select
e.employee_id,
e.first_name,
e.job_id,
e.salary,
case when salary >10000 then 'High salary' else 'Low Salary' end salary_status,
e.department_id,

d.DEPARTMENT_NAME,

d.location_id,

l.country_id,

c.country_name,

j.job_title

from employees e inner join (select department_id,department_name,location_id from departments) d

on e.department_id=d.department_id

join locations l

on d.location_id=l.location_id

join countries c

on l.country_id=c.country_id

join jobs j

on e.job_id=j.job_id

where e.job_id like  '%CLERK' and  upper(c.country_name) like '%UNITED%';
```

https://stackoverflow.com/questions/16263652/multiple-table-join-query-with-count-in-oracle-sql

```
--------------------------------------------------------------------------------

--Cross Join or Cartesian product

select * from departments;

select

e.employee_id,

e.first_name,

e.email,
e.SALARY,
e.department_id,
d.DEPARTMENT_NAME,
d.LOCATION_ID
from employees e, departments d
--where e.department_id=d.department_id;
```

107*27=2889

```
--------------------------------------------------------------------------------
```

self Join

```
select * from employees;
```

```
select
e1.employee_id,
e1.first_name,
'reports to',
e2.employee_id,
e2.first_name
from employees e1,employees e2
where e1.manager_id=e2.employee_id order by e1.employee_id;
```

```
--------------------------------------------------------------------
```

--Non-Equi Join

```
select
e.employee_id,
e.first_name,
e.email,
e.SALARY,
e.department_id,
d.DEPARTMENT_NAME,
d.LOCATION_ID
from employees e, departments d
where e.department_id!=d.department_id;
```

Join with USING clause

```
SELECT employee_id,first_name,job_title,
      department_name, city
FROM employees
INNER JOIN departments  USING (department_id)
INNER JOIN locations USING (location_id)
INNER JOIN jobs USING (job_id)
ORDER BY 1;
```
